Car DVD BT Module User Manual

V0.1 – Feb 15, 2008

19, Innovation First Road • Science Park • Hsin-Chu • Taiwan 300 • R.O.C.

Tel: 886-3-578-6005 Fax: 886-3-578-4418 Web: www.sunplus.com

Important Notice

SUNPLUS TECHNOLOGY CO. reserves the right to change this documentation without prior notice. Information provided by SUNPLUS TECHNOLOGY CO. is believed to be accurate and reliable. However, SUNPLUS TECHNOLOGY CO. makes no warranty for any errors which may appear in this document. Contact SUNPLUS TECHNOLOGY CO. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by SUNPLUS TECHNOLOGY CO. for any infringement of patent or other rights of third parties which may result from its use. In addition, SUNPLUS products are not authorized for use as critical components in life support systems or aviation systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Sunplus.

Revision History

Revision	Date	By	Remark	Page Number(s)
V 0.1	2008/02/15	xujf	First	All

Table of Content

1 Introduction

1.1 General Description

This guide describes the functionality and user API of BT Module for SPHE8202T CAR DVD system.

2 Data Struct Introduce

2.1 数据存储结构：

Struct:

```
typedef struct
{
    UINT16   xStart;
    UINT16   yStart;
    UINT16   xSize;
    UINT16   ySize;
} BT_pos;
typedef struct
{
    BT_pos   position;
    const BYTE *bmp;
    BYTE   str_id;
    void (*Touch_Function)(UINT32 Touch_x , UINT32 Touch_y,UINT32 gEvent_ID);
} BT_menu_object;
const BT_menu_object BT_menu_pos[];
const BT_menu_object BT_dial_page_button[];
const BT_menu_object BT_dial_vpp_button[];
const BT_menu_object BT_Vol_button[];
const BT_menu_object BT_keybd_button[];

typedef struct{
    BYTE BTStatus;                          //the BT macro state,=NONE;
    BYTE sParameter[MAX_DIAL_NUM];    //this var save the info which user wanna.
    BYTE ucLength;                          //follow up the length of sParameter[],=0;
    TX_BUF tx_buf;                    //structed the HBCP raw transfered data package
```

```
        HBCP_RX_INFO gHBCP_Rx;              //structed the HBCP raw received data package
        BYTE gHBCP_Rx_Buf[RX_BUF_MAX_SIZE];       //store the the HBCP data package
        BYTE rxBuf[RX_BUF_MAX_SIZE];
        BYTE gTxSeq;                              //=0;      // tx sequencial number
        //BYTE gRxSeq;                            //=0;// rx sequencial number
        #ifdef NEW_RX_PROCESS
            BYTE parse_rxBuf[RX_BUF_MAX_SIZE];   //mapping with rxBuf[],we copy rxBuf to
            UINT16 rxIn;                         //the rxBuf data write-point writed by UART
            UINT16 rxOut;   //the rxBuf data read-point read by user(that's copy to parse_rxBuf)
            //UINT16 rxSize;                     //no use now
            UINT16 gRx_chk_sum;                  //sum of all rx data
            UINT16 gRx_data_sum;                 //the sum data sent from BlueCore
        #else
            BYTE uart_receive_start;            //receive one HBCP data package start flag,=0;
            BYTE uart_receive_end;              //receive one HBCP data package end flag,=0;
            BYTE uart_receive_num;              //how many bytes haved been read from one
            received HBCP data package end flag,=0;
        #endif
        BYTE sync_send;                         //is the flag of sending sync cmd to BT core
        BYTE sync_OK;                           //is the flag of BT core send sync OK to host
        UINT16 cmd_delay_send_timer;       //use this timer to send some cmd to BT core

//some BT core data useful for user,like paired device name,incoming call NO.,address list...//
        BYTE    pair_device_name[MAX_DEVICE_NUM][MAX_DEVICE_NAME];    //store    the
        paired device name
        BYTE connect_device_name[MAX_DEVICE_NAME];
        BYTE BT_hfp_volume;
        BYTE BT_av_volume;
            //user system/ui/other var//
        BYTE BT_cmd;                            //BT user command
        #ifdef POWER_ON_BT_ON
            BYTE BT_init;     //mainly used to identify the BT core has been power on or not,=0;
        #endif
        #ifdef BT_AUTO_SEARCH     //08/01/16 temp added for auto search,remove soon
            BYTE BT_init_timer;
        #endif
        BYTE call_coming_in_other_state;            //a flag just like the var name,=0;
        BYTE BT_ui_state;                       //a flag just like the var name,=BT_STATE_IDLE;
        BYTE BT_ui_state_old;                   //save the prev BT_ui_state
        //#ifdef SIMPLE_OSD_DEMO_BT
        BYTE BT_state_str[MAX_STATE_STR_NUM];     //
        //#endif
        BYTE force_refresh; //force system/ui refresh the BT state,it afford for thought,maybe
    removed soon
        BYTE pair_device_num;

        BYTE latest_call_num;
        BYTE miss_call_num;
        BYTE command_ready;
        BYTE show_title_ready;
        BYTE connect_by_ag;

        }BLUETOOTH_VAR;

      EXTERN BLUETOOTH_VAR bt_var;
```

Attribute Introduce

BT_menu_object position：
存储当前可用 ICON 的响应范围。
const BYTE *bmp：
存储当前可用 ICON 上的 BMP DATA，有些 ICON 是没有的则此项为空。
str_id：
为当期可用 ICON 上的显示字，有些 ICON 上是没有的则此项为空。
void (*Touch_Function)(UINT32 Touch_x , UINT32 Touch_y,UINT32 gEvent_ID)：
当前 ICON 对应的功能执行函数。

EXTERN BLUETOOTH_VAR bt_var;变量的参数介绍：

BTStatus // 南牙模组当前所出的状态。
sParameter[MAX_DIAL_NUM]; // 存储的是当前需要的一些数据，如电话号码，姓名等。
ucLength // 统计 sParameter[]中有效数据的长
tx_buf // 存储的是串口发送的数据信息。
gHBCP_Rx // 存储的是串口收到的数据头信息
gHBCP_Rx_Buf // 存储的是串口收到的有效数据。
rxBuf // 存储的是串口收到的原始数据。
gTxSeq // 被发送数据的编号
parse_rxBuf //需要解析的数据
rxIn/ rxOut // 向串口进行读写的状态标志
pair_device_name[][] // 存储南牙模组中已匹配的 Device 名。
connect_device_name[] // 存储南牙已经建立链接的 Device 名。
BT_hfp_volume // 电话音量
BT_cmd //向南牙发送的命令
其余的是一些显示控制需要的变量。

结构体变量 BLUETOOTH_VAR bt_var 为数据与控制相关的变量组。

BT_menu_object 为 UI 使用的结构体。

3 BT Function Explain

3.1 BT UI 说明

(图一)

(图二)

(图三)

(图四)

以上四图是所有的 BT 操作界面。图一为主界面，图二为显示已匹配的设备，图三是本机的匹配码设置，图四为接听/拨打电话界面。

3.2 BT 功能说明

BT 功能有：

1.读取最近通话记录。

2.读取未接来电记录。

3.耳机音量调节。

4.重拨功能。

5.来电自动接听功能。

6.有 DTMF 功能。

7.可以设置本机匹配码。

8.有自动连接功能。

4 How To Add A New BT Item

4.1 Introduce

BT 模块是比较简单的，大部分工作都由 BT 模组完成了，我们只要向 BT 模组发命令就可以了，SPHE8202X 这端主要就是设计 UI。下面向你介绍如何增加一个 BT item.

4.2 Create UI mode

1.首先要知道你需要建立的新的 UI 界面有哪些 TOUCH Button。

2.建立这些 TOUCH Button 的管理数组，如 const BT_menu_object BT_menu_pos[];
const BT_menu_object BT_dial_page_button[];等

3.建立创建该界面的绘制函数如 show_XX_ui（)。在这个函数中既绘制了其基本画面，也创建了该界面下的各个 TOUCH Button 的命令函数。

4.一些界面绘制需要的 ICON,BMP 等数据请放在#include "BT_UI_icon.inc"文件中。

5 User Interface Function

5.1 show_dial_ui

画拨号界面

Function	Required Header
show_dial_ui	BT_UI_Draw.h

void show_dial_ui(void)

Parameter

NULL

Return Values

NULL

5.2 show_BT_menu_ui

画 BT 主 UI 界面

Function	Required Header
show_BT_menu_ui	BT_UI_Draw.h

void show_BT_menu_ui(void)

Parameter

NULL

Return Values

NULL

5.3 show_install_ui

画匹配码设置界面

Function	Required Header
show_install_ui	BT_UI_Draw.h

void show_install_ui(void)

Parameter

NULL

Return values

NULL

5.4 show_device_ui

画已匹配设备列表界面

Function	Required Header
show_device_ui	BT_UI_Draw.h

void show_device_ui(void)

Parameter

NULL

Return values

NULL

5.5 show_BT_menu_content

基于 BT_menu_object 结构的 UI BUTTON 绘制函数，它完成 Button 的背景，BMP 图片绘制和 Button 上的文字显示绘制。

Function	Required Header
show_BT_menu_content	

void show_BT_menu_content(const BT_menu_object *pos_menu)

Parameter

const BT_menu_object *pos_menu：提供绘制该 Button 需要的信息

Return values

NULL

5.6 show_BT_button

基于 BT_menu_object 结构的 UI BUTTON 绘制函数，它完成 Button 的背景和 Button 上的文字显示绘制。该函数没有绘制 BMP 图片的功能。

Function	Required Header
show_BT_button	

void show_BT_button(const BT_menu_object *pos_menu)

Parameter

const BT_menu_object *pos_menu：提供绘制该 Button 需要的信息

Return values

NULL

5.7 show_BT_Phone_vpp_button

它完成 Button 的背景绘制。该函数只绘制 VPP BMP 图片。

Function	Required Header
show_BT_Phone_vpp_button	

void show_BT_Phone_vpp_button (const BT_menu_object *pos_menu)

Parameter

const BT_menu_object *pos_menu：提供绘制该 Button 需要的信息

Return values

NULL

5.8 create_BT_keybd_button

它完成 Keyboard Button 的背景绘制。它完成 Button 的背景，Button 上的文字显示绘制。

Function	Required Header
create_BT_keybd_button	

void create_BT_keybd_button (const BT_menu_object *pos_menu)

Parameter

const BT_menu_object *pos_menu：提供绘制该 Button 需要的信息

Return values

NULL

5.9 BT_key

执行 BT 命令

Function	Required Header
BT_key	BT_global.h

void BT_key(BYTE cmd)；

Parameter

BYTE cmd :传递执行的命令。

Return values

NULL

5.10 HandleCommand

此函数执行上层发送的命令

Function	Required Header
HandleCommand	BT_global.h

void HandleCommand(UINT16 command)；

Parameter

UINT16 command :传递执行的命令。

Return values

NULL

5.11 UartDrv_Rx

接收由串口得到的数据

Function	Required Header
UartDrv_Rx	BT_global.h

void UartDrv_Rx(BYTE *in_buf, UINT16 buf_len, UINT16 *in_len);

Parameter

*in_buf：存放接收到的数据 Buffer.

buf_len: 为 Buffer 最大空间。

*in_len: 记录次此收到的数据长度。

Return values

NULL

5.12 HBCP_Receive

解析接收到的数据.

Function	Required Header
HBCP_Receive	BT_HBCP.h

void HBCP_Receive(BYTE ucLen, unsigned char *pStr);

Parameter

BYTE ucLen：当前接收到的数据长度.

unsigned char *pStr ：当前接收到的数据.

Return values

NULL

5.13 HBCP_RcvPacket

解析去掉包头的有用数据包.在该函数中最终解析得到具体的数据。

Function	Required Header
HBCP_RcvPacket	BT_global.h

void HBCP_RcvPacket(void)

Parameter

NULL

Return values

NULL

5.14 HBCP_Send_Command

发送命令函数，由 SPHE8202X 向蓝牙模组发送命令。

Function	Required Header

HBCP_Send_Command	BT_HBCP.h

void HBCP_Send_Command(BYTE ucCmd, int ucLen, BYTE *parameter)

Parameter

BYTE ucCmd：需要蓝牙模组执行的命令。

BYTE *parameter：有些命令是有带参数的，该形参就是为这些命令设的。

int ucLen：表示该命令所带参数的长度。

Return values

NULL

6 BT Command list

```
#define HBCP_CMD_HSHF_ENTER_PAIR                ((unsigned char)0x01)
#define HBCP_CMD_HSHF_EXIT_PAIR                 ((unsigned char)0x02)
#define HBCP_CMD_HSHF_SET_PIN                   ((unsigned char)0x03)
#define HBCP_CMD_HSHF_CALL_ACCEPT               ((unsigned char)0x05)
#define HBCP_CMD_HSHF_CALL_REJECT               ((unsigned char)0x06)
#define HBCP_CMD_HSHF_VOL_UP                    ((unsigned char)0x07)
#define HBCP_CMD_HSHF_VOL_DOWN                  ((unsigned char)0x08)
#define HBCP_CMD_HSHF_DIAL_LAST                 ((unsigned char)0x09)
#define HBCP_CMD_HSHF_DIAL_NUM                  ((unsigned char)0x0A)
#define HBCP_CMD_HSHF_DIAL_VOICE                ((unsigned char)0x0B)
#define HBCP_CMD_HSHF_CALL_END                  ((unsigned char)0x0C)
#define HBCP_CMD_HSHF_GET_LIST                  ((unsigned char)0x0D)
#define HBCP_CMD_HSHF_SET_STORAGE               ((unsigned char)0x11)
#define HBCP_CMD_HSHF_SEND_DTMF                 ((unsigned char)0x12)
#define HBCP_CMD_HSHF_INQUIRY           ((unsigned char)0x16)/*sxg 20070105*/
#define HBCP_CMD_HSHF_CANCEL_INQUIRY((unsigned char)0x17)/*sxg 20070105*/
#define HBCP_CMD_HSHF_GET_STATUS                ((unsigned char)0x18)
#define HBCP_CMD_HSHF_GET_CONNECTED_DEVICE_NAME ((unsigned char)0x19)
```

```
#define HBCP_CMD_SYNC                          ((unsigned char)0x25)
#define HBCP_CMD_HSHF_BLUECORE_ON_OFF          ((unsigned char)0x27)
#define HBCP_CMD_HSHF_CONNECT                  ((unsigned char)0x2A)
#define HBCP_CMD_HSHF_DISCONNECT               ((unsigned char)0x2B)
#define HBCP_CMD_HSHF_CALL_TRANS_TO_AG         ((unsigned char)0x2C)
#define HBCP_CMD_HSHF_CALL_TRANS_TO_HF         ((unsigned char)0x2D)
#define HBCP_CMD_HSHF_DELETE_PAIR              ((unsigned char)0x31)
#define HBCP_CMD_HSHF_CONNECT_AG_BY_ADDR       ((unsigned char)0x13)
#define HBCP_CMD_HSHF_CALL_SWAP                ((unsigned char)0x10)
#define HBCP_CMD_HSHF_REL_ACT_ACC_OTH          ((unsigned char)0x2E)
#define HBCP_CMD_HSHF_REL_HOL_REJ_WAIT         ((unsigned char)0x2F)
#define HBCP_CMD_HSHF_ADD_HOLD_CALL            ((unsigned char)0x30)
#define HBCP_CMD_HSHF_MUTE                     ((unsigned char)0x3A)
#define HBCP_CMD_HSHF_GET_PAIR_DEV_NAME        ((unsigned char)0x3C)
#define HBCP_CMD_HSHF_RESTORE_FACTORY          ((unsigned char)0x3D)
#define HBCP_CMD_AV_STOP_AUDIO                 ((unsigned char)0x44)
#define HBCP_CMD_AV_PLAY_PAUSE_AUDIO           ((unsigned char)0x45)
#define HBCP_CMD_AV_PREVIOUS_AUDIO             ((unsigned char)0x46)
#define HBCP_CMD_AV_NEXT_AUDIO                 ((unsigned char)0x47)
#define HBCP_CMD_SET_BC_NAME        ((unsigned char)0x50)
#define HBCP_CMD_SET_BC_FNAME((unsigned char)0x51)
#define HBCP_CMD_SET_BC_TITLE       ((unsigned char)0x52)
#define HBCP_CMD_SET_BC_BIRTH       ((unsigned char)0x53)
#define HBCP_CMD_SET_BC_ADDR        ((unsigned char)0x54)
#define HBCP_CMD_SET_BC_ORG         ((unsigned char)0x55)
#define HBCP_CMD_SET_BC_URL         ((unsigned char)0x56)
#define HBCP_CMD_SET_BC_TEL         ((unsigned char)0x57)
#define HBCP_CMD_SET_BC_MOBILE      ((unsigned char)0x58)
#define HBCP_CMD_SET_BC_FAX         ((unsigned char)0x59)
#define HBCP_CMD_SET_BC_EMAIL       ((unsigned char)0x5a)
```

```
#define HBCP_CMD_SET_BC_TZ        ((unsigned char)0x5b)
#define HBCP_CMD_SET_BC_REV       ((unsigned char)0x5c)
#define HBCP_CMD_CONNECT_SPP_BY_ADDR      ((unsigned char)0x60)
#define HBCP_CMD_CONNECT_LAST_SPP ((unsigned char)0x61)
#define HBCP_CMD_DISCONNECT_SPP   ((unsigned char)0x62)
#define HBCP_CMD_SPP_DATA         ((unsigned char)0x63)
```

7 Limits and Suggestions

Some limits exist in this Setup module. We list them as following.